USN | | | | | | | | | | |

Internal Assessment Test 1 – Sept. 2019

Sub:	Operating System					Sub Code:	17 EC 553	Branch:	ECE/TCE	
Date:	-9-19	Duration:	90 min's	Max Marks:	50	Sem / Sec:	5 – A B C D		OBE	

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1	Define Operating System. What are the goals of Operation systems? Explain	[10]	CO1	L1
2	Explain key features of Batch processing and Multi programming OS with neat diagram and explain their advantages and Disadvantages	[10]	CO1	L2
3	Explain with neat sketch the view of OS on process / Process Environment & PCB structure	[10]	CO2	L2
4	Define process state. With neat sketch explain the process fundamental state transition diagram or process life cycle.	[10]	CO2	L2
5	Explain key features of Time sharing processing and Real Time OS with neat diagram and explain their advantages and Disadvantages.	[10]	CO1	L2
6	Explain following terms 1) Preemption 2) Remote Procedure Call (RPC) 3) Dispatching 4) Context Switching 5) OS Modes – user mode & privilege mode	[10]	CO2	L2
7	What are the advantages of threads over process? Explain kernel level threads.	[10]	CO2	L1

USN | | | | | | | | | | |

Internal Assessment Test 1 – Sept. 2019

Sub:	Operating System					Sub Code:	17 EC 553	Branch:	ECE/TCE	
Date:	-9-19	Duration:	90 min's	Max Marks:	50	Sem / Sec:	5 – A B C D		OBE	

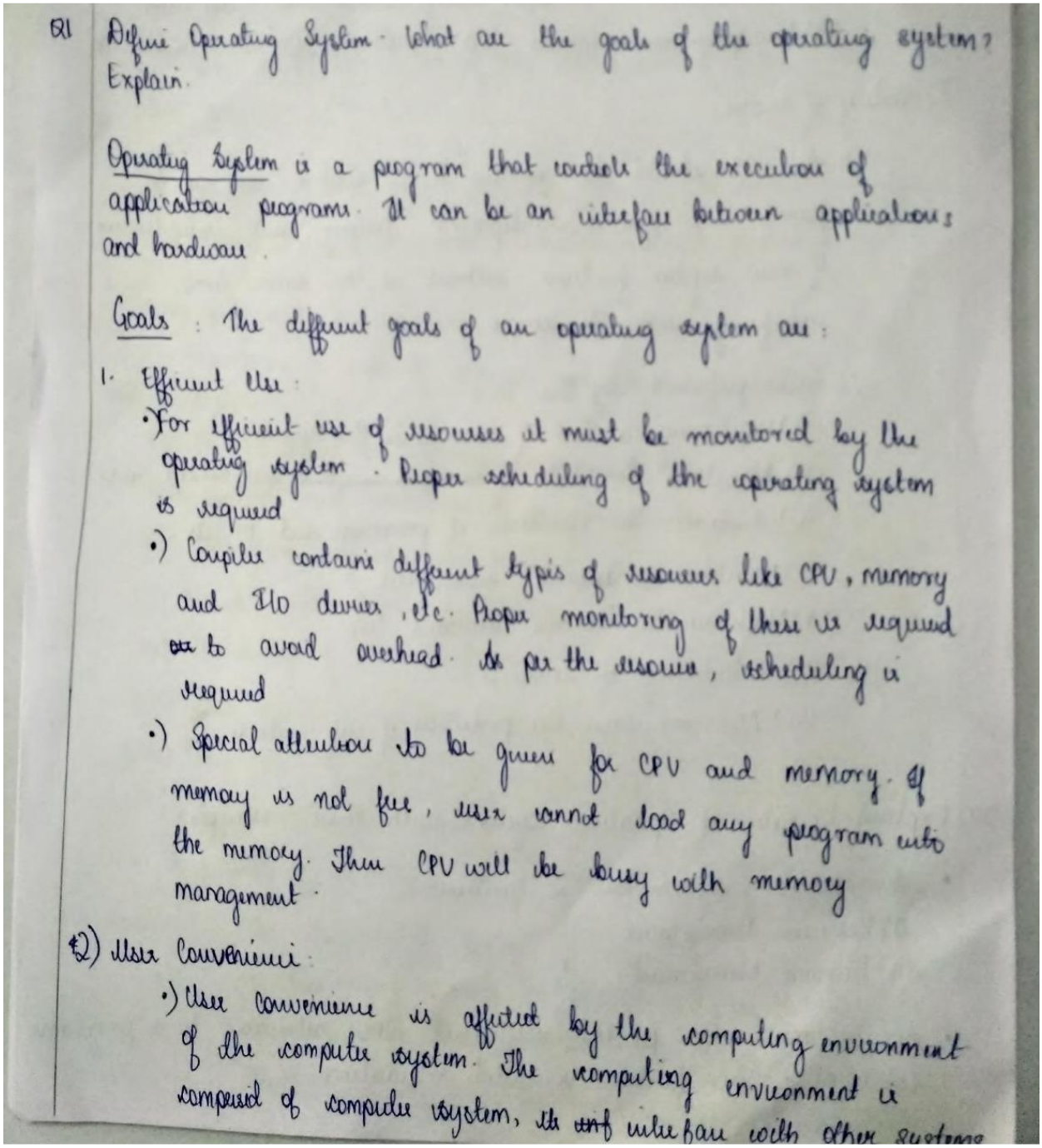
Answer any FIVE FULL Questions

		MARKS	CO	RBT
1	Define Operating System. What are the goals of Operation systems? Explain	[10]	CO1	L1
2	Explain key features of Batch processing and Multi programming OS with neat diagram and explain their advantages and Disadvantages	[10]	CO1	L2
3	Explain with neat sketch the view of OS on process / Process Environment & PCB structure	[10]	CO2	L2
4	Define process state. With neat sketch explain the process fundamental state transition diagram or process life cycle.	[10]	CO2	L2
5	Explain key features of Time sharing processing and Real Time OS with neat diagram and explain their advantages and Disadvantages.	[10]	CO1	L2
6	Explain following terms 1) Preemption 2) Remote Procedure Call (RPC) 3) Dispatching 4) Context Switching 5) OS Modes – user mode & privilege mode	[10]	CO2	L2
7	What are the advantages of threads over process? Explain kernel level threads.	[10]	CO2	L1

Operating System ----Internal Assessment Test 1 – Sept. 2019

Solution:

1)



Q1 Differ Operating System. What are the goals of the operating system? Explain.

Operating System is a program that controls the execution of application programs. It can be an interface between application and hardware.

Goals : The different goals of an operating system are:

1. Efficient Use:
   - For efficient use of resources it must be monitored by the operating system. Proper scheduling of the operating system is required
   - Computer contains different types of resources like CPU, memory and IO devices, etc. Proper monitoring of these is required to avoid overhead. As per the resource, scheduling is required
   - Special attention to be given for CPU and memory. If memory is not free, user cannot load any program into the memory. Then CPU will be busy with memory management.

2) User Convenient:
   - User convenience is affected by the computing environment of the computer system. The computing environment is composed of computer system, its interface with other systems

and nature of computations performed by the user.

- Computer architecture and user change the computing environment of the system. Following factors are considered while considering user convenience: (i) Good Service (ii) Ease of Use (iii) New programming models (iv) Evolution (v) User friendly OS

5) Ability of Evolve.

- An OS should be constructed in such a way as to permit the effective development, testing and introduction of new system features without at the same time interfering with the service.
- Tasks performed by the OS:
  - i) Maintaining a list of resources in the system
  - ii) Maintain the list of ~~resources in the~~ authorized users
  - iii) Initiate the execution of programs and process.
  - iv) Maintain ~~as~~ resource usage list.
  - v) Maintain the resource allocated list.
  - vi) Scheduling of resources
  - vii) Also maintain the protection of information.

2)

Q4 Explain key features of batch processing and multiprogramming OS with neat diagram and explain their advantages and disadvantages

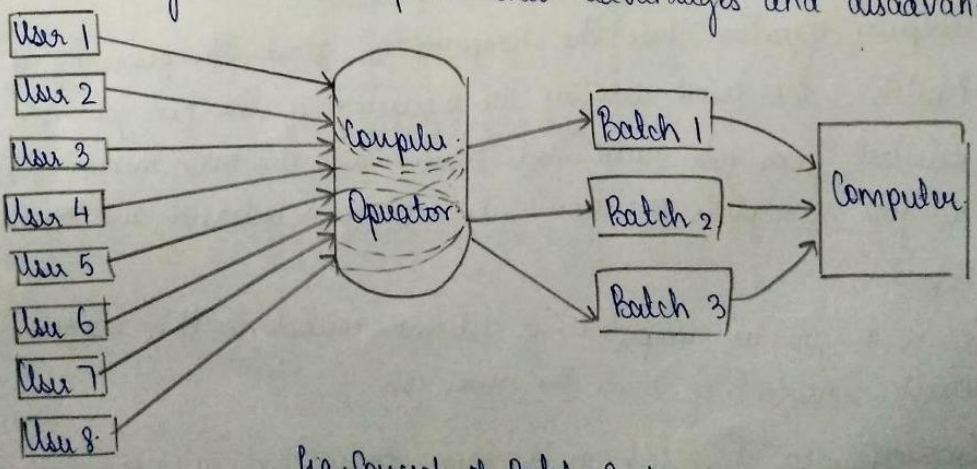


fig: Concept of Batch System

- Batch is a collection of jobs, called a batch. Batch is a sequence of user jobs.
- Job is a predefined sequence of commands programs and data that are combined into a single unit.
- Each job in a batch is independent of other jobs in a batch
- Jobs with similar needs were batched together to speed up processing
- Card readers and tape drives are the input devices in batch system. Output devices are tape drives, card punches and line printer.
- Primary function of batch system is to execute the jobs in a batch one after another without requiring the operators intervention.
- Some computers have a serial system where a list of instructions are carried out one after another.
- Batch monitor is used to implement batch processing system. Batch monitor is also called Kernel. Kernel resides in one part of the computer main memory.
- Batch monitor controls the sequence of events. Main memory store the batch monitor and users program and data.
- Computer operator gives the command to start the processing of a batch, the kernel sets up the processing of the first job. Job was selected from job queue and loaded into the main memory. When a job completes execution its memory is released and o/p is copied.
- When a job is completed, it returns control to the monitor which immediately reads the next job
- Spooling uses the disk as a large buffer for outputting data to

printer is and other devices

ADVANTAGES

1. More much of the work of the operator to the ~~computer~~ computer
2. Increased performance since it was possible for job to start as soon as the previous job finished.

DISADVANTAGES

1. Turn around time can be large from user standpoint
2. Program debugging is difficult.
3. There was possibility of entering jobs in infinite loop
4. A job could corrupt the monitor, thus affecting pending jobs

MULTIPROGRAMMING OS

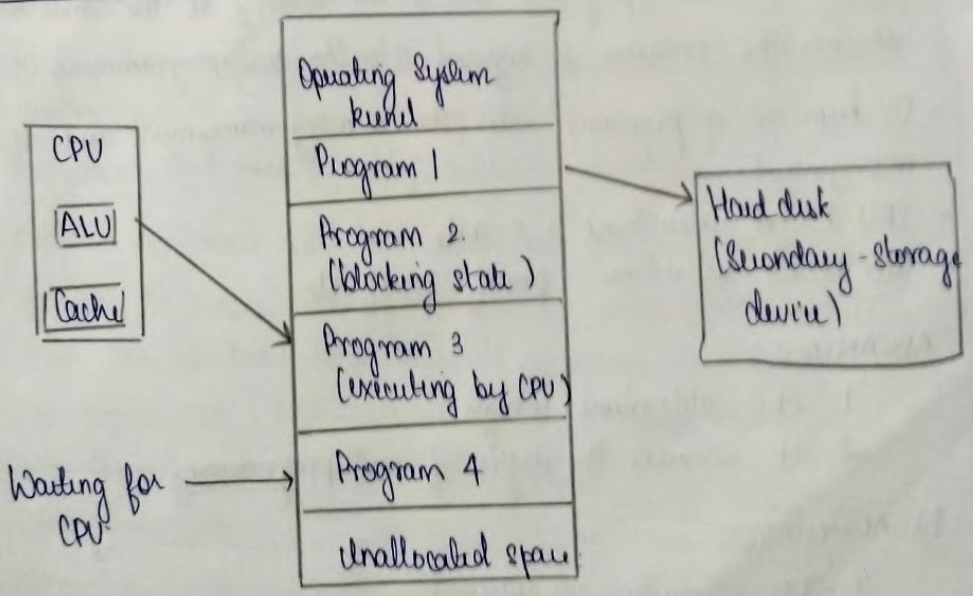


fig. Multiprogramming OS

- At any time either the CPU (or) I/O devices is idle in a batch systems. To keep CPU busy, more than one program/job must be loaded for execution. So multiprogramming increases the CPU utilization.
- Resource management is the main aim of multiprogramming OS, file system, command processor, I/O control system and transient area are the essential components of a single user operating system. Multiprogramming OS divides the transient area to store the multiple ~~languages~~ programs and provides resource management to the OS.
- A program in execution is called a 'process', 'job' or a 'task'
- When 2 or more programs are in the memory at the same time, sharing the processor is suffered to the multiprogramming OS.
- OS keeps no. of programs into the memory management and I/O management
- CPU bound instruction : c=a+b
- I/O bound instruction : printf, scanf, etc.

ADVANTAGES

1. CPU utilization is high.
2. It increases the degree of multiprogramming

DISADVANTAGES

1. CPU scheduling is required
2. Memory management is required

3)

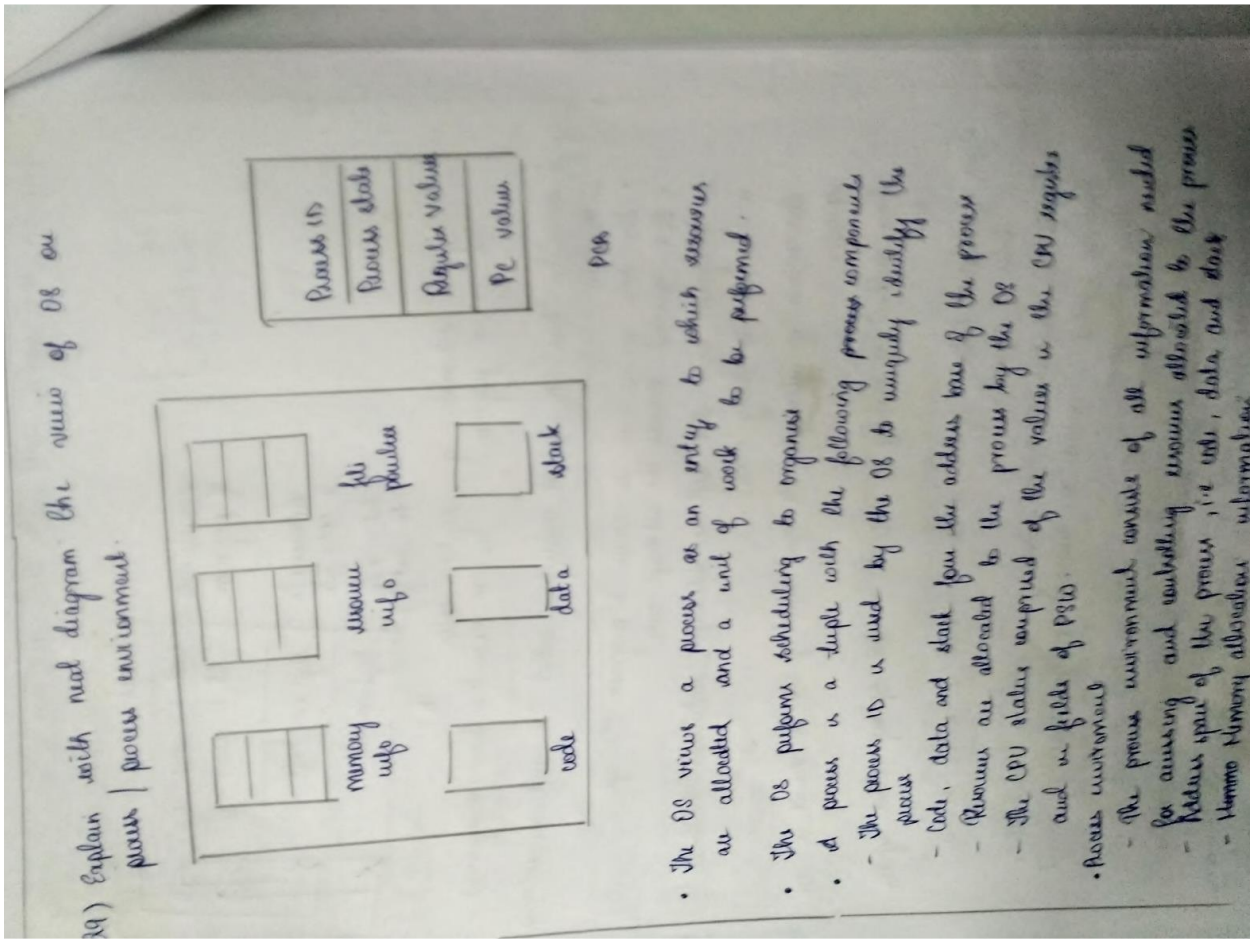


Figure Kernel's view of a process.

PCB field	Contents
Process id	The unique id assigned to the process at its creation.
Parent, child ids	These ids are used for process synchronization, typically for a process to check if a child process has terminated.
Priority	The priority is typically a numeric value. A process is assigned a priority at its creation. The kernel may change the priority dynamically depending on the nature of the process (whether CPU-bound or I/O-bound), its age, and the resources consumed by it (typically CPU time).
Process state	The current state of the process.
PSW	This is a snapshot, i.e., an image, of the PSW when the process last got blocked or was preempted. Loading this snapshot back into the PSW would resume operation of the process. (See Fig. 2.2 for fields of the PSW.)
GPRs	Contents of the general-purpose registers when the process last got blocked or was preempted.
Event information	For a process in the *blocked* state, this field contains information concerning the event for which the process is waiting.
Signal information	Information concerning locations of signal handlers (see Section 5.2.6).
PCB pointer	This field is used to form a list of PCBs for scheduling purposes.

4)

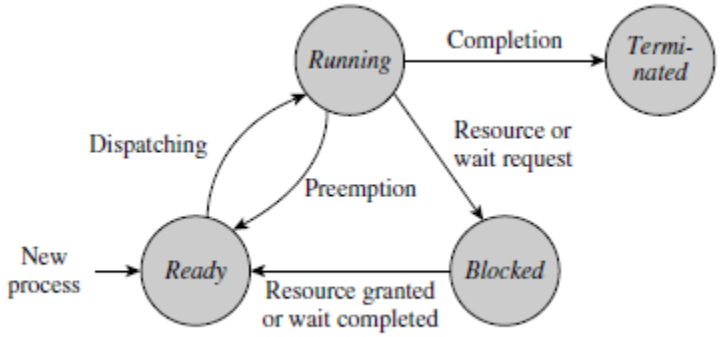
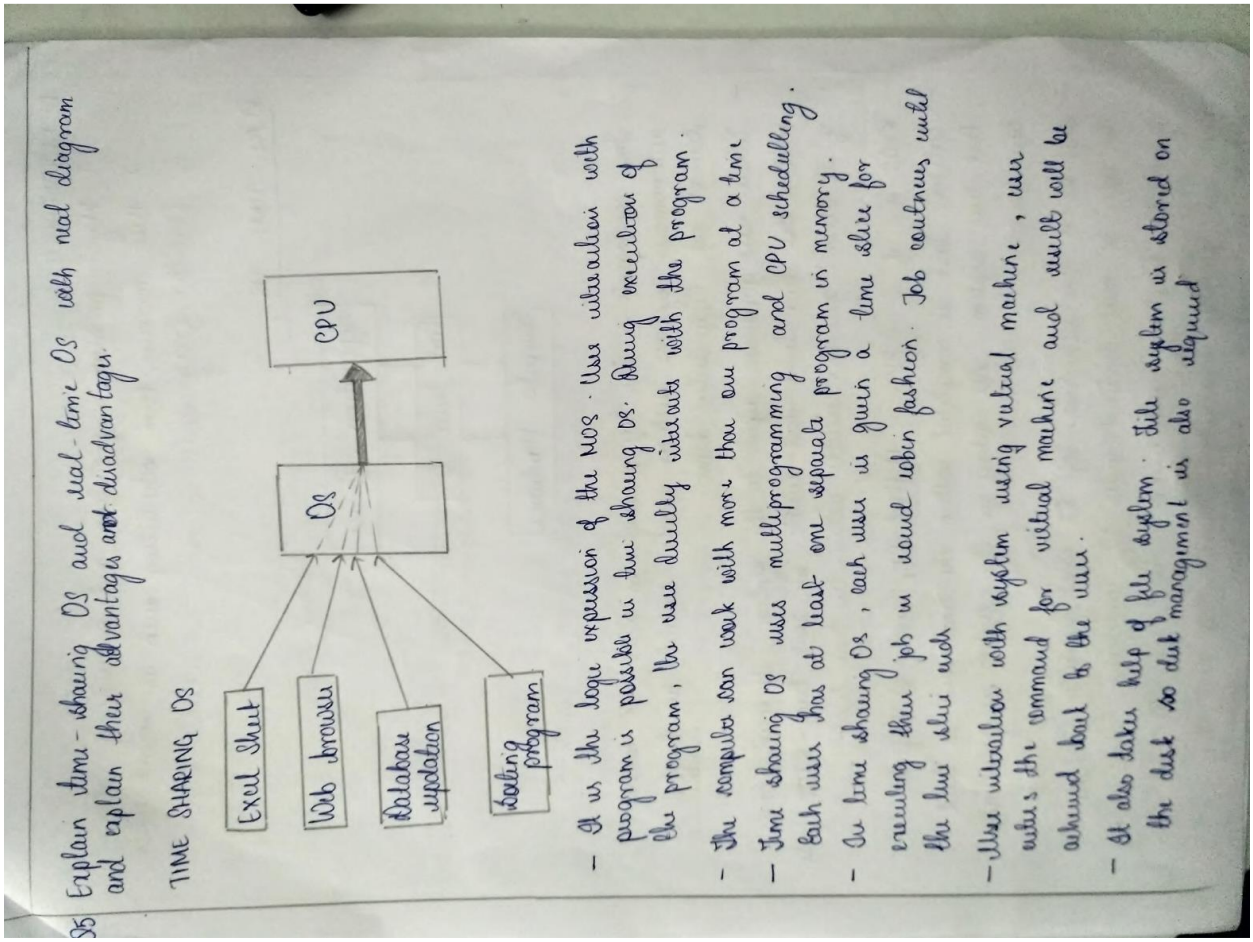


Figure Fundamental state transitions for a process.

Table Causes of Fundamental State Transitions for a Process

State transition	Description
ready → *running*	The process is dispatched. The CPU begins or resumes execution of its instructions.
blocked → *ready*	A request made by the process is granted or an event for which it was waiting occurs.
running → *ready*	The process is preempted because the kernel decides to schedule some other process. This transition occurs either because a higher-priority process becomes *ready*, or because the time slice of the process elapses.
running → *blocked*	The process in operation makes a system call to indicate that it wishes to wait until some resource request made by it is granted, or until a specific event occurs in the system. Five major causes of blocking are: • Process requests an I/O operation • Process requests a resource • Process wishes to wait for a specified interval of time • Process waits for a message from another process • Process waits for some action by another process.
running → *terminated*	Execution of the program is completed. Five primary reasons for process termination are: • *Self-termination:* The process in operation either completes its task or realizes that it cannot operate meaningfully and makes a "terminate me" system call. Examples of the latter condition are incorrect or inconsistent data, or inability to access data in a desired manner, e.g., incorrect file access privileges. • *Termination by a parent:* A process makes a "terminate P_i" system call to terminate a child process P_i, when it finds that execution of the child process is no longer necessary or meaningful.

5)



6) a) The OS takes away the CPU from a program after it has executed for the specified period of time, and gives it to another program. This action is called *preemption*. A program that loses the CPU because of preemption is put back into the list of programs waiting to execute on the CPU.
The scheduling policy employed by an OS can influence both efficient use of the CPU and user service. If a program is preempted after it has executed for only a short period of time, the overhead of scheduling actions would be high because of frequent preemption. However, each program would suffer only a short delay before it gets an opportunity to use the CPU, which would result in good user service. If preemption is performed after a program has executed for a longer period of time, scheduling overhead would be lesser but programs would suffer longer delays, so user service would be poorer.

b) A process calls a procedure that is located in a different computer system. The RPC is analogous to a procedure or function call in a programming language, except that the OS passes parameters to the remote procedure over the network and returns its results over the network.

c) *Dispatching:* Setting up access to resources of the scheduled process and loading its saved CPU state in the CPU to begin or resume its operation.

d) *Context save:* Saving CPU state and information concerning resources of the process whose operation is interrupted.

e) Kernel and User Modes of CPU Operation The CPU can operate in two modes, called *user mode* and *kernel mode*. The CPU can execute certain instructions only when it is in the kernel mode. These instructions, called *privileged instructions*, implement special operations whose execution by user programs would interfere with the functioning of the OS or activities of other user programs; e.g., an instruction that changes contents of the *memory protection information* (MPI) field of the PSW could be used to undermine memory protection in the system. The OS puts the CPU in kernel mode when it is executing instructions in the kernel, so that the kernel can execute special operations, and puts it in user mode when a user program is in execution, so that the user program cannot interfere with the OS or other user programs. We assume the *mode* (M) field of the PSW to be a single-bit field that contains a 0 when the CPU is in kernel mode and a 1 when it is in user mode.

8)

Advantages of Threads over Processes

Advantage	Explanation
Lower overhead of creation and switching	Thread state consists only of the state of a computation. Resource allocation state and communication state are not a part of the thread state, so creation of threads and switching between them incurs a lower overhead.
More efficient communication	Threads of a process can communicate with one another through shared data, thus avoiding the overhead of system calls for communication.
Simplification of design	Use of threads can simplify design and coding of applications that service requests concurrently.